Motion Synthesis By Example

Alexis Lamouret and Michiel van de Panne
Department of Computer Science
University of Toronto

Abstract

A technique is proposed for creating new animations from a set of representative example motions stored in a motion database. Animations are created by cutting-and-pasting together the example motion segments as required. Motion segments are selected based upon how well they fit into a desired motion and are then automatically tailored for a precise fit. Various fundamental problems associated with the use of motion databases are outlined. A prototype implementation is used to validate the proposed concepts and to explore possible solutions to the aforementioned problems.

Keywords: computer animation, motion capture, motion reuse, motion database

1 Introduction

Building tools for character animation has long been a difficult business. The problem of building a tool to help animators in their job is ill-defined in many ways. Just how does an animator communicate a set of desired motion characteristics to a software tool? How does the tool inform the animator of possible constraints or compromises in achieving a desired motion? This paper investigates the possibility of generating motions-by-example: the animator provides the tool with a series of sample motions for the character. This *motion database* can then be used to generate further motions of the same style. Such an animation tool is aimed precisely at automating the predictable aspects of motions, whatever they may be.

Like much of science, many approaches to animation have been deconstructionist, seeking to build models of motion by breaking it down into its constituent components. This has yielded mechanical models to which the laws of physics are applied and which must be controlled using a simulated musculature[1, 5, 13, 7, 12, 9, 10]. The idea behind such *physics-based animation* is to automatically constrain animated motions to be physically plausible. It has since been noted by many that this is a necessary, but not sufficient condition for producing *natural* motion. In order to obtain desired natural motions, we additionally need to choose control actions which are also 'natural'. This, unfortunately, has proved to be a difficult problem to solve.

Recent advances in motion capture technology has made it the animation method of choice in a variety of situations. By allowing acted human movements to be directly mapped onto an animated character, the creation of a desired motion is direct and instantaneous. Recent work has focussed on how to edit and manipulate the motion data in order to make it more adaptable and reusable[14, 11, 4]. These efforts provide methods of post-processing the motion capture data. The animator can use a series of semi-automated tools to alter and tailor a chosen segment of motion capture data to meet particular desired constraints.

Kinematic models of motion capture the essence of a particular type of motion (e.g., walking) and attempt to parameterize it in a useful way[2, 3, 6, 8]. For example, kinematic models of walking typically let the animator directly control the speed, stride-length, and direction of the walking gait. The motion models are typically constructed and tuned by extracting important characteristics from real motion data. The synthesis of kinematic motion models is a fairly complex procedure, making it non-trivial to develop new motion models.

In order to arrive at a more flexible model of character motion, we propose to use a large set of representative motions together with techniques for tailoring these motions to make them fit new situations. In effect, the proposed technique consists of sampling a phenomena and being able to reconstruct new instances of the phenomena using these sample points. In our case, the phenomena represents a type of character motion and the new sample points represent this type of character motion executed in a new environment.

A brief example will serve to illustrate the fundamentals of the proposed technique. Consider a human walk across variable terrain. Sample walk sequences across variable terrain are stored in a motion database and used as the basis for constructing motion across a new instance of variable terrain. For each stride, the database is queried for the stride which best represents the previous action under similar circumstances. The criteria for the best-fit involves a match of the state of the body, as well as a match of the terrain. Once found, the best-fit step is adapted to precisely fit its current use and the entire process is repeated for the next stride.

While building a motion database may in reality be infeasible for a variety of reasons, the preliminary results are sufficient to show its promise. Unlike previous techniques for altering motion-capture data, the proposed technique performs most of the work of motion synthesis through the intelligent choice of a similar motion among a set of examples.

2 Motion Databases: Problems to Overcome

A motion model seeks to represent the following function in as general a way as possible:

$$x \times e \times s \rightarrow m$$

where x is the state of the character, e is the state of the environment, s is the style of motion (e.g., big steps or bent knees), and m is a motion history of finite duration (e.g., one stride for a walking gait). A motion database provides a set of discrete samples of this function, $\{m_1 ... m_n\}$, which serve as prototypical examples in the construction of new motion instances. Any example motion m_i can be adapted to fit new circumstances, yielding a tailored motion, m_i^*.

Many issues arise in the construction and use of a motion database. The following list is a brief exploration of some of them.

1. *What duration should the stored sample motions have?*

   The database consists of a series of example *motion primitives*. Motions of a cyclic nature, such as walking or hopping, have an obvious choice of granularity, namely a single cycle. Other types of motion may not be so easy to break into constituent motion primitives.

2. *What distance metric should be used to define the fit of a motion?*

   A best-fit motion primitive chosen to be appended to a current motion should satisfy several possibly-conflicting characteristics. The starting state (position and velocity) should be a good match with the current state of the character. Similarly, the state of the environment should also be a good match. Lastly, if the style of motion is also made a parameter, it needs to be part of the matching criteria. The distance metric must weight these factors appropriately. It should be efficient to extract the best-fit motion from the set of motion examples.

3. *How can the best-fit motion primitive be adapted to precisely fit the current sitation?*

   The required C^0 continuity of the initial state (position and velocity) as well as environmental constraints prevent a direct cut-and-paste of a best-fit motion into the final motion. Recent techniques do not directly address the problem of how to modify a motion to satisfy constraints involving interaction with the environment. The reconstruction technique should preserve the original motion characteristics as much as possible.

4. *How can a motion database be kept to a minimal size?*

   There are many possible approaches to reduce the size of a motion database. All of them involve restricting redundancy in the data. One potential method consists of only adding sample motions to a database when they pass a redundancy test. Another would be to use compression techniques which take advantage of existing similarities between motion samples. Alternatively, the existing database could be resampled, the existing motion samples being replaced by a new, minimal number of optimally-placed motion prototypes. Lastly, the separability of motions can be exploited. For example, perhaps upper and lower body motions for humans can be carried out largely independently of each other, thereby allowing them to be individually matched and spliced together to produce a final motion. Statistical correlations can perhaps serve as an indicator of which joint movements can be treated independently.

   Another intriguing possibility is that of being able to take a single continuous stream of motion capture data, perhaps several hours worth collected from a person going about everyday activities, and to produce a reduced, minimal size motion database from this data.

5. *How can ambiguous actions in the database be dealt with?*

   If for a given situation (state, environment, and style of motion) multiple motions exist in the database, what should be done? The database can be considered to be inconsistent, in effect storing multiple responses to a single query. All of the responses are in principle satisfactory, allowing for an arbitrary choice among them.

6. *How can higher-level planning be carried out using the motion primitives?*

   Many actions require advance planning and anticipation. For example, carrying out a jump may require effecting preparations for the jump several steps in advance in order to build the necessary momentum. While a motion database helps define the immediate capabilities of a character, it does not provide a means of ordering the motion primitives to achieve a given goal. Planning techniques thus need to be developed which can use a motion database to achieve higher-level goals.

3 A Prototype System

As an initial investigation into the feasibility of the proposed method, we have implemented a motion database for 'Luxo', a planar 3-link articulated figure capable of a dynamic hopping behaviour. The 5 degrees of freedom of Luxo are shown in Figure 1. While it is easy to dismiss this figure as having a trivial number of degrees of freedom, its movements over variable terrain are dynamic and unstable, making it surprisingly difficult to animate using physics-based methods. By analogy, many human motions in sports are considered difficult because of their dynamic nature and the importance of the timing of the movements, rather than the number of degrees of freedom used in executing the movement.

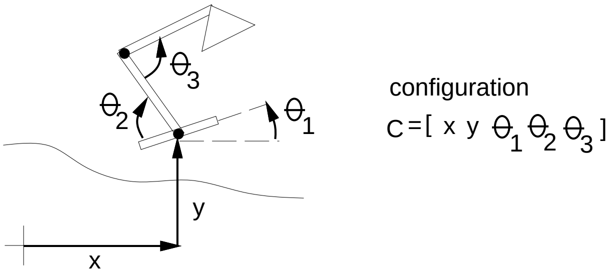

Figure 1: Luxo

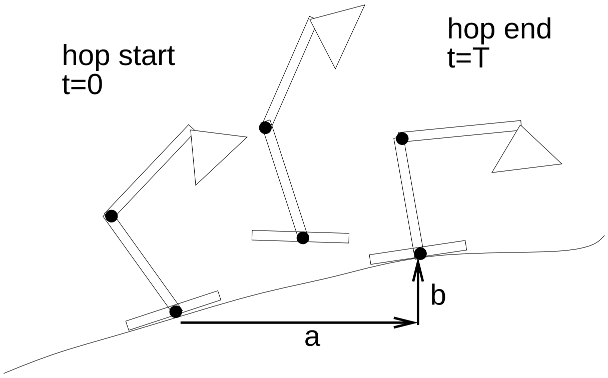

Figure 2: Measuring hop distance (a) and hop height (b).

3.1 The Motion Database

The motion samples for Luxo are generated using physics-based animation. Five different types of control actions are defined, each of which causes a different style of hop to be executed. The total number of possible motions is much larger than the limited set of five control choices because the motions are also very sensitive to (a) the initial conditions for the hop, and (b)the state of the environment. A decision-tree technique is used to plan the control actions as Luxo travels across variable terrain. This technique simulates the effect of control decisions several steps into the future, much like a game-playing algorithm and produces successful plans for Luxo leaping across variable terrain. The necessity of carrying out multiple simulations makes generating a physics-based animation considerably more expensive than the proposed synthesis-by-example technique.

The choice of primitive for Luxo is straightforward – it consists of a single hop. The start and end of motion hops are defined by the controller responsible for generating the motions. The result of a single sequence of physics-based animation is thus broken into multiple motion primitives, each of which serves as an entry in the motion database.

Being able to conduct an efficient search for the best-fit motion is important and therefore the motion samples are stored in a structured fashion. In the case of Luxo, jumps are classified according to the length, a, and height b of the jump, measured as shown in Figure 2. An example of the resulting discretized classification of the hops for Luxo in a motion database is shown in Figure 3.

During the search for the best-fit motion, only samples within cells intersecting the terrain are considered for a best-fit calculation, as shown in Figure 4. This quickly reduces the number of motion samples to search according the present 'state' of the

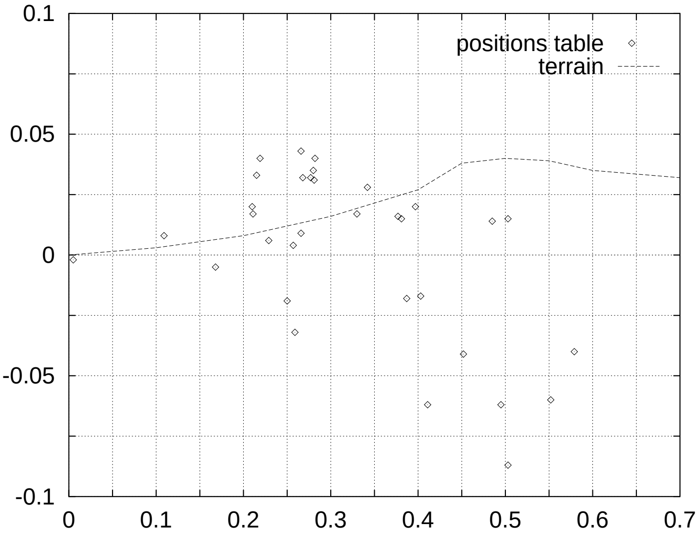

Figure 3: A motion database with hops classified according to jump distance (x-axis), and jump height (y-axis). Samples close to the shown terrain instance might be suitable candidates to be used for the next hop.

Figure 4: Using the terrain to restrict the candidate motions

environment. In the prototype implementation, issues of compression, redundant motion samples, and higher-level planning are not dealt with at present.

3.2 The Distance Metric

Deciding on the 'fit' of a sample motion requires looking at both the continuity of the motion as well as its suitability given the terrain. Further user specifications might also come into effect, such as a preferred hop style. A database query retrieves the motion corresponding to the following minimum distance:

$$d_{min} = min_{\{m_i\}}(d_{state} + k_1 d_{env} + k_2 d_{user})$$

where

d_{state}	measures the compatibility of the initial state
d_{env}	measures the compatibility of the environment (terrain)
d_{user}	measures the compatibility with particular user specifications

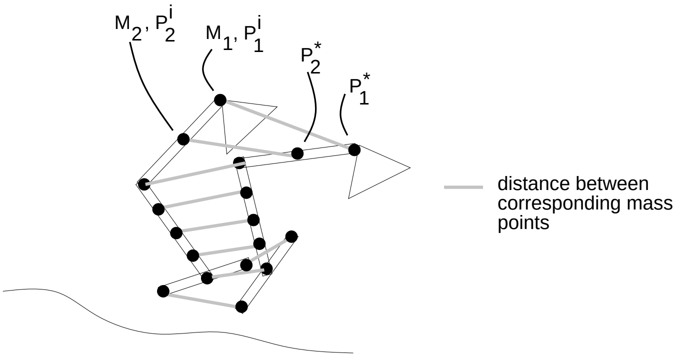

Figure 5: The mass distance metric

k_1, k_2 adjust the relative importance of the terms

Various metrics are possible to measure the distance between two states. A straightforward approach is to calculate an L^2 norm on the state vector:

$$d_{state} = \sum_{j=1}^{N} c_j (x_j^i - x_j^*)^2 .$$

Unfortunately, this is also a misguided approach, as the choice of state vector is not unique and furthermore does bad job of measuring the possible correspondence of two states. An improved scheme is the mass-distance method proposed in Figure 5:

$$d_{state} = \sum_{m} M_m \|p_m^i - p_m^*\|^2$$

Here, a point-mass distribution is created to approximate the real mass distribution of the articulated figure. The square of the dislocation of these point masses, weighed proportionately by their mass is used as a distance metric which is independent of the choice of state vector.

One item yet to be addressed is that of matching velocities. It is possible to have two creatures be in an identical position but having different velocities. We choose to ignore this problem at present.

The d_{env} term is calculated using the square of the distance between the true landing point for a sample hop and the closest landing point in the given environment. An example of the necessaary type of translation is shown in Figure 7, which will be discussed in further detail shortly.

3.3 Adapting the Best-fit Motion

Once a best-fit motion primitive has been found, it needs to be adapted to be continuous with the previous motion as well as to precisely enforce any necessary constraints imposed by the environment. The adapting process is best illustrated by viewing the trajectory in the configuration space[1], as shown in Figure 6. The specifics of an adapation of a Luxo hop are shown in Figure 7.

First, the adaptation necessary to make the starting state, $x(0)$, match the final position of the existing motion, x_f, is determined: $\Delta x_1 = x_f - x(0)$.

Second, an adaptation to make the final position of the motion properly match the given terrain constraints is determined: $\Delta x_2 = x_c - x(T)$, where T is the duration of the

---

[1] The configuration space defines all possible positions of the degrees of freedom

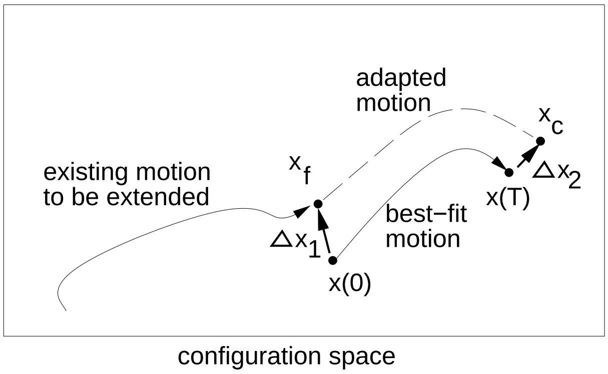

Figure 6: Adapting the best-fit motion.

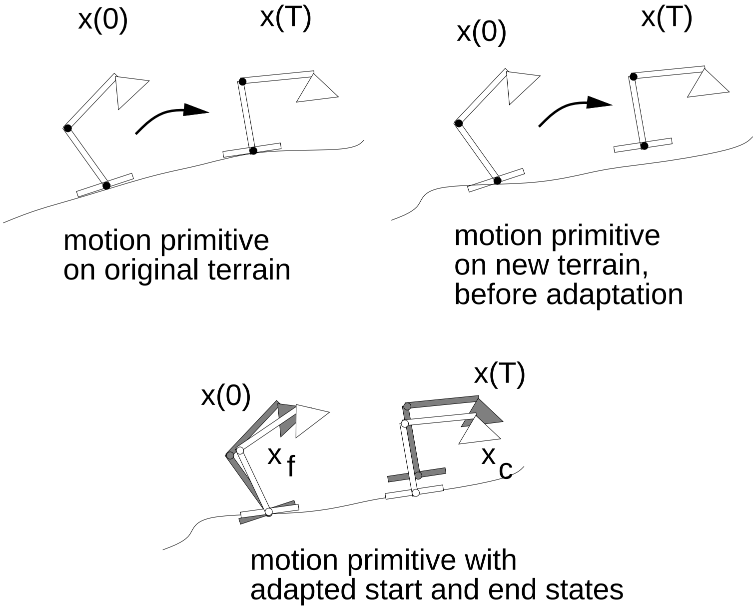

Figure 7: Adapting a Luxo jump.

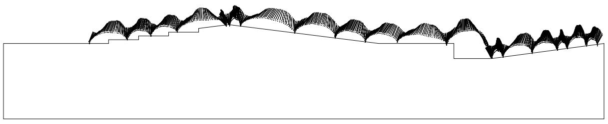

Figure 8: Motion Dataset 1

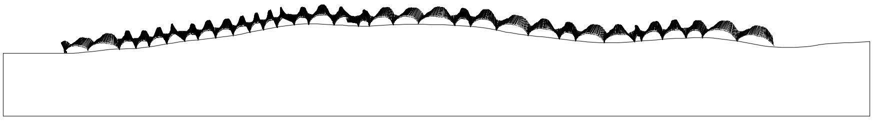

Figure 9: Motion Dataset 2

motion primitive, $x(t)$. For Luxo, this involves finding the minimal displacement such that the terrain contact configuration matches that found in the sample motion. This includes ensuring the contact of specific points with the terrain as well as maintaining the same relative angle of the base with respect to the terrain. The adaptations carried out at $x(0)$ and $x(T)$ should by their nature be specific to the character being animated, as the types of constraints which need to be satisfied will depend on the character.

The final adapted motion is defined by gradually phasing in or out the adaptations over the time-duration T:

$$\alpha = t/T$$

$$x'(t) = x(t) + (1-\alpha)\Delta x_1 + \alpha\Delta x_2$$

In the above discussion, $x(t)$ and $x'(t)$ refer only to the configuration over time, and thus not the true state including velocities. Smoother 'ease-in' and 'ease-out' functions are also easy to implement.

3.4 Results

The prototype system has thus far been evaluated using relatively small sets of example motions. Figure 8 and 9 show motions synthesized directly using physics-based animation. In both cases, only the motion of Luxo's 'leg' link is shown for the purposes of clarity. The motion shown in Figure 9 will be used as the basis of our motion database in the further examples to be shown.

Results of the animation-by-example technique are illustrated in Figures 10–12. Figure 10 illustrates a motion on the terrain of Figure 8 constructed using the motion examples from Figure 9. Figure 11 is constructed in an identical fashion, except that a user-term has been added to the matching function which encourages the use of small hops. Figure 12 illustrates motion on a new instance of terrain, once again using the motion examples extracted from Figure 9.

While subjective judgement of the resulting motions reveals them to be missing some of the fluidity of the original motions, the motions nevertheless appear quite plausible to the eye. Figure 13 shows a detail of the original, physics-based animation of Figure 8. A detailed view of the synthetic motion across the same terrain is shown in Figure 14. The synthetic motion is in this case somewhat smoother than the actual motion.

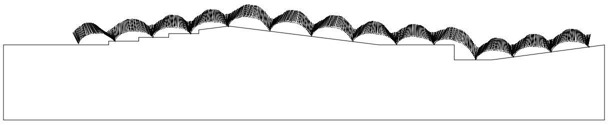

Figure 10: Output on terrain 1 with dataset 2, big hops

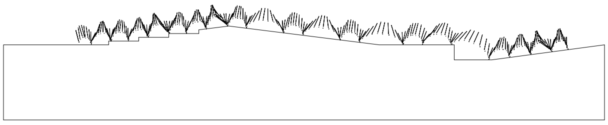

Figure 11: Output on terrain 1 with dataset 2, small hops

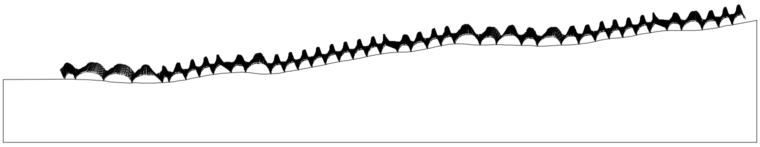

Figure 12: Output on terrain 3 with dataset 2

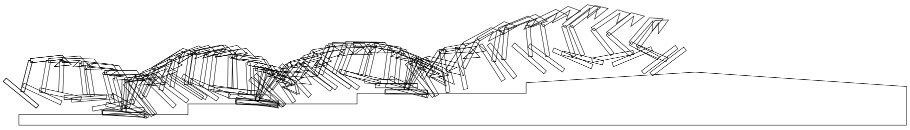

Figure 13: Detail of original motion (dataset 1).

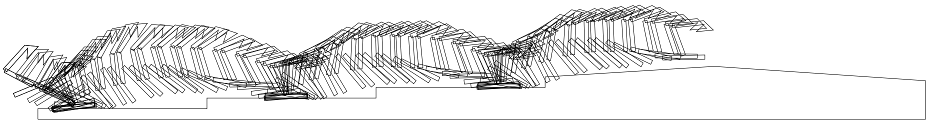

Figure 14: Detail of synthetic motion (dataset 2 applied to terrain 1).

4 Conclusions

The idea of directly modelling motion has long existed in the form of procedural animation. Similarly, the idea of adapting existing motion data to fit new situations has also recently seen considerable interest. We have speculated that it is possible to build animation systems which can 'animate by example'. The proposed framework is one of keeping a representative sample of possible motions in a motion database and being able to recall a best-fit motion for a new situation. The best-fit example motion is then precisely adapted to the new situation by applying small corrections.

A variety of subproblems of the animation-by-example technique were examined. Many of the problems to be solved are related to those of machine learning, where the goal is to learn a compact representation of a function by correctly generalizing from a series of examples (supervised learning). The subproblem of adapting a motion for a precise fit is specific to animation. Although the adaptation problem is relatively simple for our chosen examples, it will require the use of character-specific inverse kinematics for more complex figures.

Being able to perform higher-level motion planning using a motion database remains a problem of considerable interest. One could imagine extracting higher-level motion characteristics from examples. Alternatively, an accelerated planning technique could operate by composing together the primitives found in a motion database.

In order to keep a motion database to a reasonable size, a variety of compression strategies are possible. By extracting correlations between components for a movement, it should be possible to recognize, for example, that head motion is largely independent of leg motion during walking and should therefore be treated separately. An ideal system would be able to extract a compact, irreducible set of example motions given a large pool of motion data.

In conclusion, this workshop paper proposes that the idea of animation-by-example is an interesting and desireable feature for animation tools of the future. It is hoped that the ideas presented here will inspire further discussion at the workshop.

References

[1] N. I. Badler, B. Barsky, and D. Zeltzer. *Making Them Move.* Morgan Kaufmann Publishers Inc., 1991.

[2] R. Boulic, N. M. Thalmann, and D. Thalmann. A global human walking model with real-time kinematic personification. *The Visual Computer*, 6:344–358, 1990.

[3] A. Bruderlin and T. W. Calvert. Interactive animation of personalized human locomotion. *Proceedings of Graphics Interface*, pages 17–23, 1993.

[4] A. Bruderlin and L. Williams. Motion signal processing. *Proceedings of Siggaph '95, ACM Computer Graphics*, pages 97–104, 1995.

[5] J. K. Hodgins et al. Animating human athletics. *Proceedings of SIGGRAPH '95, ACM Computer Graphics*, pages 71–78, 1995.

[6] M. Girard. Interactive design of computer-animated legged animal motion. *IEEE Comptuer Graphics and Applications*, 7(6):39–51, June 1987.

[7] R. Grzeszczuk and D. Terzopoulos. Automated learning of muscle-actuated locomotion through control abstraction. *Proceedings of SIGGRAPH '95, ACM Computer Graphics*, pages 63–70, 1995.

[8] H. Ko and N. I. Badler. Straight line walking animation based on kinematic generalization that preserves the original characteristics. *Proceedings of Graphics Interface '93*, pages 9–16, 1993.

[9] J. T. Ngo and J. Marks. Spacetime constraints revisited. *Proceedings of SIGGRAPH '93*, pages 343–350, 1993.

[10] K. Sims. Evolving virtual creatures. *Proceedings of SIGGRAPH '94, ACM Computer Graphics*, pages 15–22, 1994.

[11] M. Unuma, K. Anjyo, and R. Takeuchi. Fourier principles for emotion-based human figure animation. *Proceedings of SIGGRAPH '95, ACM Computer Graphics*, pages 91–96, 1995.

[12] M. van de Panne and E. Fiume. Sensor-actuator networks. *Proceedings of SIGGRAPH '93*, pages 335–342, 1993.

[13] M. van de Panne, R. Kim, and E. Fiume. Virtual wind-up toys for animation. *Proceedings of Graphics Interface '94*, pages 208–215, 1994.

[14] A. Witkin and Z. Popovi'c. Motion warping. *Proceedings of SIGGRAPH '95*, pages 105–107, 1995.